

全脳アーキテクチャ・イニシアティブ

第2回 WBAレクチャー

ハンズオン

2021.6.13



小野寺由樹

荒川直哉

本ハンズオンでは、

作業記憶をテーマとした第5回WBAハッカソンで提供するサンプルコードについて

1. 開発環境をインストール
2. 課題の体験やエージェントの訓練を動かしてみる

サンプルコード: [https://github.com/wbap/WM\\_Hackathon](https://github.com/wbap/WM_Hackathon)

# 1. 開発環境をインストール

今回、説明する開発環境



docker の開発環境を  
使用します。



Mac

Anaconda(Miniconda)  
で開発環境をインストー  
ルします。

## 事前準備の確認(Windows)



①WSL2(Windows 10 用 Windows Subsystem for Linux)のインストール

<https://docs.microsoft.com/ja-jp/windows/wsl/install-win10#manual-installation-steps>

手動インストールの手順から、手順 1~5 を実施してください。

②Docker Desktop for Windows のインストール

<https://www.docker.com/get-started>

Docker Desktopの[Download for Windows]をクリックしてインストーラをダウンロードします。インストーラを実行してください。

Docker のWSLディストリビューションを確認: `wsl -l -v`

## 事前準備の確認(Windows)



### ③Git for Windows のインストール

<https://gitforwindows.org/>

[Download]をクリックしてインストーラをダウンロードします。インストーラを実行してください。

※改行コードが Windows-style(CRLF) に変換されないようにします。

`git config -l` で確認。 `core.autocrlf=true` になっていないこと。

`true` だったら `false` に変更します。

```
git config --global core.autocrlf false
```

## 事前準備の確認 (Windows)



### ④VcXsrv Windows X Server のインストール

<https://sourceforge.net/projects/vcxsrv/>

[Download]をクリックしてインストーラをダウンロードします。インストーラを実行してください。

### ⑤Docker イメージのダウンロード

次のコマンドを実行して、Docker イメージをダウンロードしてください。

```
docker pull cerenaut/wbai_wm_hackathon:latest  
(5.5 GB)
```

## 事前準備の確認 (Mac)



### ① Anaconda または Miniconda のインストール

#### ・Anacondaの場合

<https://www.anaconda.com/products/individual#Downloads>

MacOSのインストーラをダウンロードします。インストーラを実行してください。

※どっちがいいかわからない場合は、[64-Bit Graphical Installer]をダウンロードしてください。

#### ・Minicondaの場合

<https://docs.conda.io/en/latest/miniconda.html#macosx-installers>

MacOSXのインストーラをダウンロードします。インストーラを実行してください。

※どれがいいかわからない場合は、Python 3.8の[Miniconda3 MacOSX 64-bit pkg]をダウンロードしてください。



## 事前準備の確認(Mac)



### ②Git のインストール

ターミナルで `git --version` と打ってみてください。

バージョンが表示されたら、使える状態です。

「コマンドライン・デベロッパ・ツール」のインストールを促す画面が表示されたら[インストール]をクリックしてください。

## 事前準備の確認(共通)



# Windows 10

## ソースコードのダウンロード

開発環境のディレクトリを作成して、そのディレクトリに移動します。

(例:ホームディレクトリの Dev ディレクトリ)

```
mkdir ~/Dev
```

```
cd ~/Dev
```

次のコマンドを実行して、ソースコードをダウンロードしてください。

```
git clone https://github.com/wbap/WM_Hackathon.git
```

```
git clone https://github.com/Cerenaut/cerenaut-pt-core.git
```

開発環境 Wiki もご覧ください。

[https://github.com/wbap/WM\\_Hackathon/wiki/Development-Environment](https://github.com/wbap/WM_Hackathon/wiki/Development-Environment)

## 1. Using Docker for your Development Environment

 Windows10 ⇒OK

## 2. Set up your own Development Environment

 Mac ⇒続けてセットアップします。

## 2. 課題の体験やエージェントの訓練を動かしてみる

# 遅延見本合わせ課題の体験



```
python keyboard_agent.py dm2s-v0 configs/dm2s_env.json
```

## Windows 10

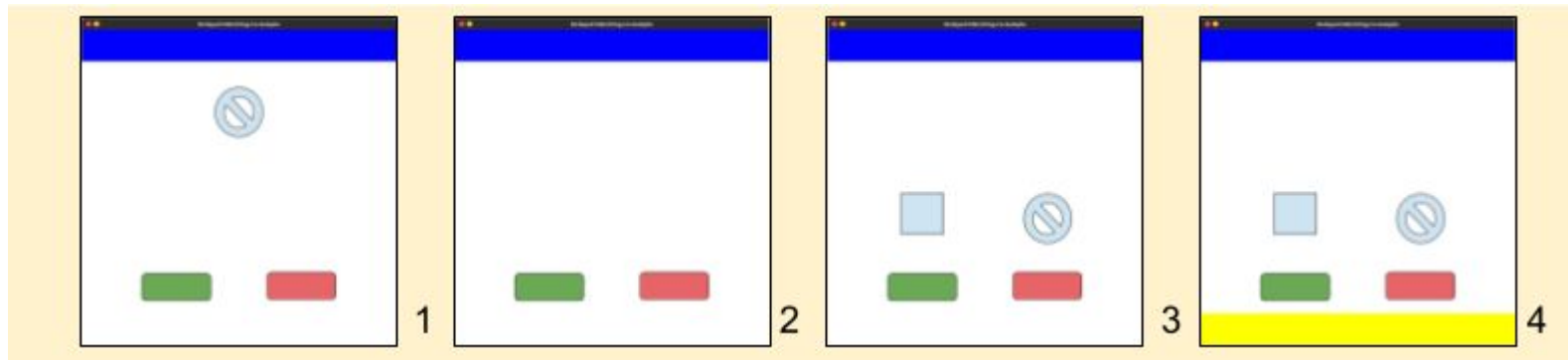
先に VcXsrv (XLaunch) を起動。※ Additional parameters for VcXsrv: `-ac`

```
deployment/run-docker.ps1 ~/Dev/WM Hackathon  
~/Dev/cerenaut-pt-core $false python keyboard_agent.py dm2s-v0  
configs/dm2s_env.json
```

※ Windowsでは“~”（チルダ）を引数に使えません。絶対パスを指定してください。

# 遅延見本合わせ課題の体験

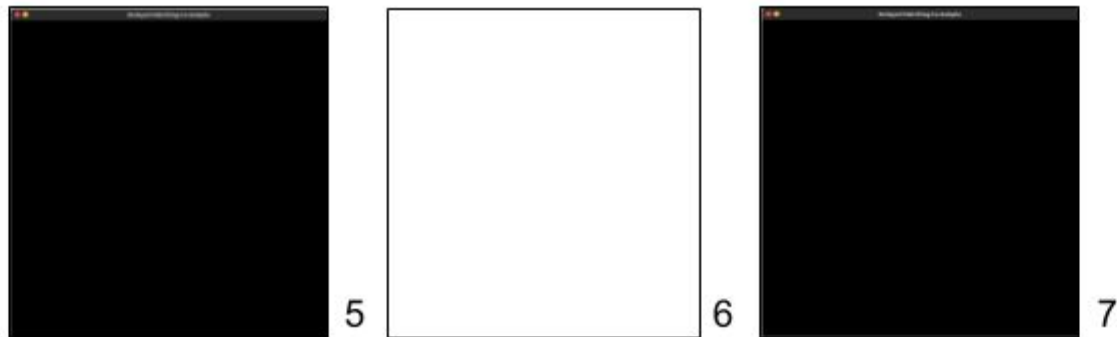
## ①ルール例示(画面上部に青いバー)



形の比較 / 色の比較 / 棒の位置の比較

# 遅延見本合わせ課題の体験

②テスト開始の合図(画面がフラッシュ)



# 遅延見本合わせ課題の体験

## ③テスト本体



解答入力： 1（左側）、2（右側）

画面下部が黄色いバー（正解）、赤いバー（不正解）



## エージェントの訓練

- 基本エージェントの訓練と評価を実行
  - 非遅延見本合わせ課題
  - 周辺視・中心視の区別がない完全視野  
(カメラの視野)
- TensorBoard でモニタリング

# 非遅延見本合わせ課題の体験



```
python keyboard_agent.py m2s-v0 configs/m2s_env.json
```

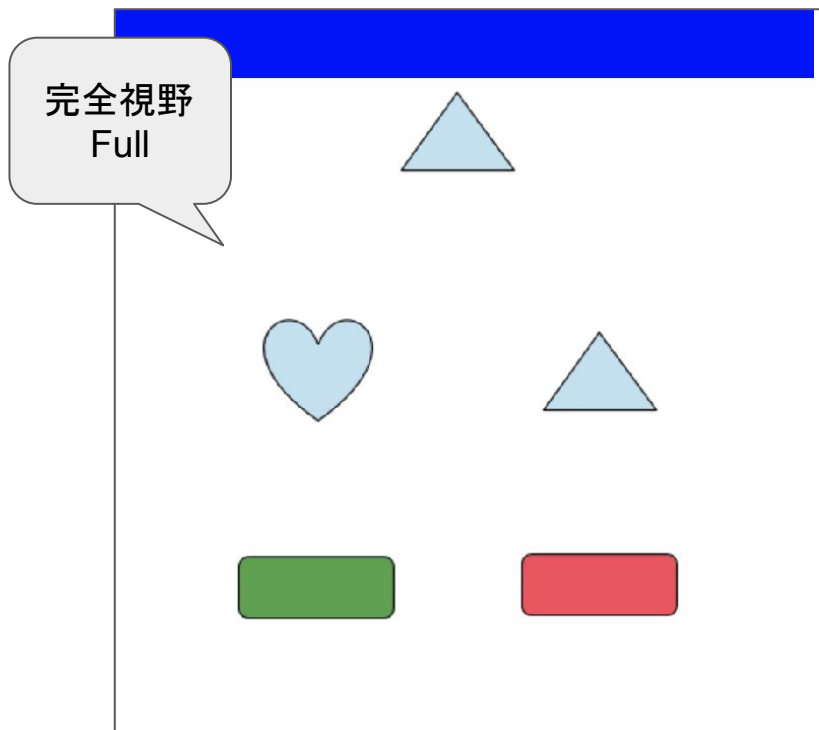
## Windows 10

先に VcXsrv (XLaunch) を起動。※ Additional parameters for VcXsrv: `-ac`

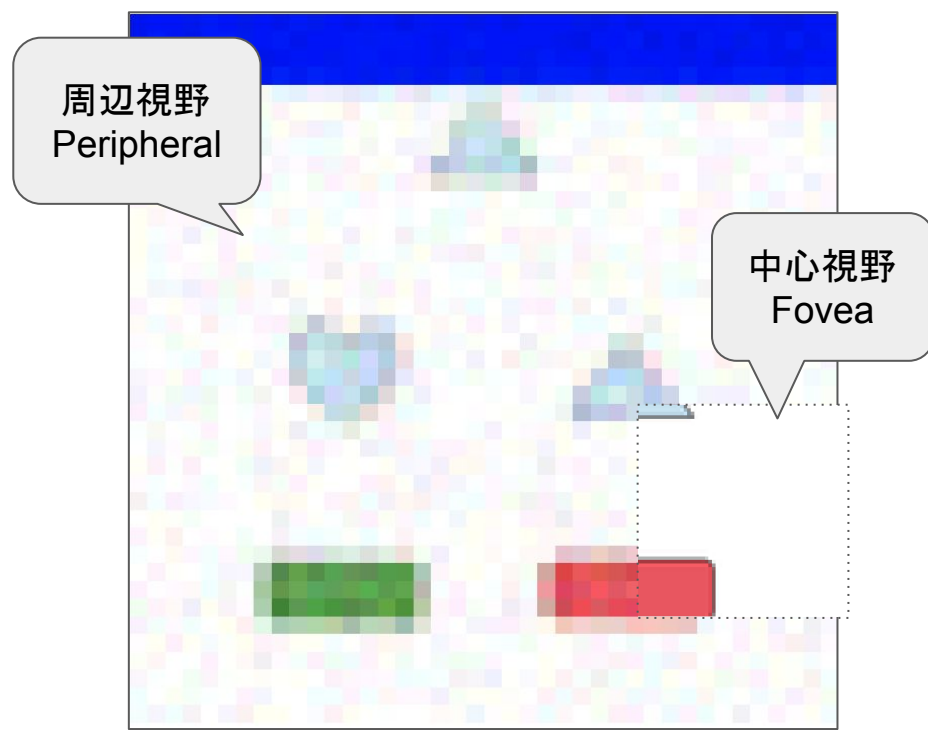
```
deployment/run-docker.ps1 ~/Dev/WM Hackathon  
~/Dev/cerenaut-pt-core $false python keyboard_agent.py m2s-v0  
configs/m2s_env.json
```

※ Windowsでは“~”（チルダ）を引数に使えません。絶対パスを指定してください。

## カメラの視野



## ヒトのような視野



# 基本エージェントの訓練と評価を実行



```
python train_agent.py m2s-v0 configs/m2s_env.json  
configs/stub_agent_env_full.json configs/stub_agent_full.json
```

## Windows 10

```
deployment/run-docker.ps1 ~/Dev/WM_Hackathon  
~/Dev/cerenaut-pt-core $false python train_agent.py m2s-v0  
configs/m2s_env.json configs/stub_agent_env_full.json  
configs/stub_agent_full.json
```

※ Windowsでは“~”（チルダ）を引数に使えません。絶対パスを指定してください。

# TensorBoard でモニタリング



Mac

ターミナルの新規ウィンドウ

```
conda activate wm_env
```

```
cd Dev/WM_Hackathon
```

```
tensorboard --logdir=.
```



Windows 10

Docker の Dashboad から  
コンテナ「wm」の CLI をクリック

```
bash
```

```
source activate wm_env
```

```
tensorboard --logdir=. --host=0.0.0.0
```

TensorBoard

URL:

<http://localhost:6006/>

より多くの情報は、

入門 Wiki をご覧ください。

[https://github.com/wbap/WM\\_Hackathon/wiki/Getting-Started](https://github.com/wbap/WM_Hackathon/wiki/Getting-Started)

以上でハンズオンは終了です。

お疲れ様でした。